EdgeCloudAI: Edge-Cloud Distributed Video Analytics

Mahshid Ghasemi, Zoran Kostic, Javad Ghaderi, Gil Zussman
Electrical Engineering, Columbia University
{mahshid.ghasemi,zk2172,jghaderi,gil.zussman}@columbia.edu

ABSTRACT

Recent advances in Visual Language Models (VLMs) have significantly enhanced video analytics. VLMs capture complex visual and textual connections. While Convolutional Neural Networks (CNNs) excel in spatial pattern recognition, VLMs provide a global context, making them ideal for tasks like complex incidents and anomaly detection. However, VLMs are much more computationally intensive, posing challenges for large-scale and real-time applications. This paper introduces EdgeCloudAI, a scalable system integrating VLMs and CNNs through edge-cloud computing. EdgeCloudAI performs initial video processing (e.g., CNN) on edge devices and offloads deeper analysis (e.g., VLM) to the cloud, optimizing resource use and reducing latency. We have deployed EdgeCloudAI on the NSF COSMOS testbed in NYC. In this demo, we will demonstrate EdgeCloudAI's performance in detecting user-defined incidents in real-time.

CCS CONCEPTS

• **Computing methodologies** → **Cooperation and coordination**;

KEYWORDS

Edge-cloud computing, Vision language models, Real-time anomaly detection

ACM Reference Format:
Mahshid Ghasemi, Zoran Kostic, Javad Ghaderi, Gil Zussman. 2024. EdgeCloudAI: Edge-Cloud Distributed Video Analytics. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3636534.3698857

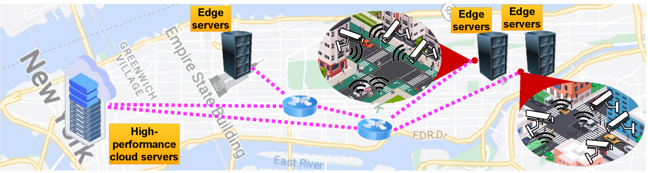
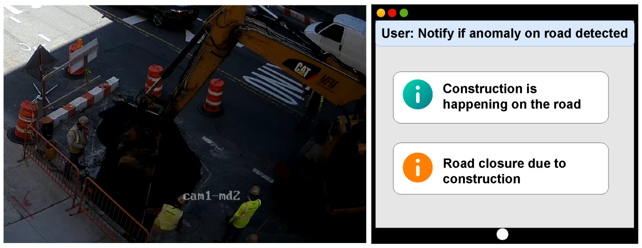


Figure 1: The NSF COSMOS testbed's geo-distributed cameras and edge-cloud servers facilitate practical evaluation [3, 19].



User: Notify if anomaly on road detected

Construction is happening on the road

Road closure due to construction

Figure 2: The left side displays the COSMOS camera's view of a road construction scene. The right side showcases the EdgeCloudAI UI for querying detected incidents and anomalies.

1 INTRODUCTION

Recent advances in Visual Language models (VLMs) [1] have significantly enhanced the potential of video analytics. Convolutional Neural networks (CNNs) have long been the backbone of various video analytics applications. CNNs are proven to be highly effective for tasks like image classification and object detection, where understanding spatial relationships is key [20, 22]. On the other hand, VLMs often incorporate Vision Transformers (ViTs) [27], which extend the transformer architecture to image processing by dividing images into patches and treating them as sequences of tokens. This allows VLMs to model complex connections between textual and visual data, potentially capturing global context more effectively than CNNs. This makes them well-suited for tasks that require understanding the semantic meaning of images, such as complex incident or anomaly detection.

Compared to CNNs, VLMs are much more computationally expensive, and they need a significant amount of power, GPU capacity, and memory. This makes their application in large-scale and real-time scenarios challenging. By integrating VLMs and CNNs through distributed edge/cloud computing, it is possible to leverage their complementary strengths. This approach enhances performance while maintaining scalability and cost-effectiveness. Cloud computing is much more computationally powerful and has more capacity than edge servers, therefore it is suitable for running

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11…$15.00
https://doi.org/10.1145/3636534.3698857

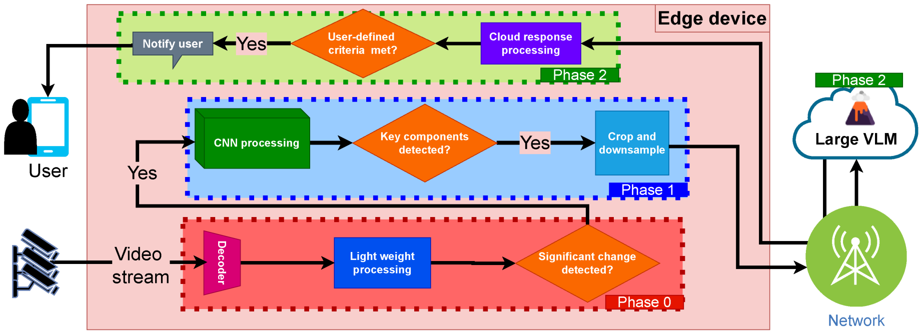


Figure 3: EdgeCloudAI pipeline architecture and its multi-phase process to detect user-defined incidents.

large VLMs. Edge servers have enough capacity for running CNNs or smaller VLMs. In this paper, we design and develop a scalable and effective system for edge-cloud integration of VLMs and CNNs. We refer to this system as *EdgeCloudAI*.

Main objective. EdgeCloudAI's main purpose is to enables scalable, low-latency, and cost-effective detection of complex incidents for applications in traffic management and transportation, industrial anomaly detection, and safety.

Key ideas. EdgeCloudAI leverages CNNs on edge to reduce the cost and latency of querying VLMs in the cloud. EdgeCloudAI performs initial processing on the edge device to determine whether to send key frames to the cloud for deeper analysis with large VLMs. It also optimizes the video's configuration, such as resolution, bitrate, and frame rate, to balance detection performance with minimal cloud costs and latency.

Several methods have been explored to enhance CNN-based video analytics performance, including selective learning [2, 8, 11, 24], online adaptation [4, 9, 10, 23], distributed inference [15, 17, 25, 26], and online filtering [16, 18, 29]. A few studies have used edge-cloud distributed computing to improve VLMs and LLMs (Large Language Models) and lower their cost and computation overhead. EdgeShard [28] partitions LLMs into multiple shards and allocates them to several edge devices and cloud servers. A dynamic token-level collaboration between small language models (SLMs) and LLMs inference on edge and cloud is proposed by [5]. In [13], split learning is used to support distributed training and inference of LLMs. Unlike previous work, **EdgeCloudAI** is a functioning real-time system–deployed in COSMOS testbed–capable of responding to users' queries. It incurs lower overhead than previous approaches by leveraging lightweight traditional computer vision methods to reduce VLMs' costs and latency while maintaining their overall accuracy.

Demo. We deployed EdgeCloudAI on NSF COSMOS testbed in New York City (NYC) (see Figure 1) [19] to evaluate its performance in realistic settings. We will demonstrate EdgeCloudAI's real-time performance using COSMOS cameras, illustrating how users can request and receive notifications for certain incidents (see Figure 2).

2 EDGECLOUDAI ARCHITECTURE

EdgeCloudAI is designed to reduce the frequency and size of cloud queries (i.e., VLM's queries) to optimize end-to-end latency and costs. Cloud providers often offer pay-as-you-go pricing, meaning you only pay per task or query rather than a fixed fee. Cloud servers may be geographically distant from data sources and connected via lower capacity links, resulting in higher network latency. In contrast, edge devices are located closer to data sources, typically connected through more reliable, higher capacity links. Edge devices are cost-effective, often available at a one-time purchase cost that amortizes over time. Considering these trade-offs, EdgeCloudAI employs the following multi-phase process to detect user-defined incidents (see Figure 3):

Phase 0: light-weight processing. The real-time video stream undergoes lightweight processing on the connected edge device to assess scene changes using methods such as background subtraction [21], optical flow [6], or a small neural net [12]. In this phase, EdgeCloudAI decides if video content changes require further analysis.

Phase 1: CNN processing. If phase 0 indicates a potential scene change, CNN models on the edge device are invoked to verify the presence of key components (e.g., objects or gestures) related to the incident of interest. If confirmed, a short video segment is prepared for transmission to the cloud, which hosts a large VLM (e.g., LLaVA.v1.6 34b [14]) for more in-depth analysis (i.e., phase 2).

Phase 2: VLM processing. If phase 2 is triggered, to enhance accuracy and ensure low latency, irrelevant parts of the video segment's frames are cropped out before sending them to the cloud. Additionally, the video segment may be downsampled based on video quality and object size to reduce data size, network latency, tokens size, and processing time. The cloud's response is subsequently processed by an SLM at the edge, ensuring that only informative notifications are sent to the user.

As an example, assume that EdeCloudAI is asked to look for construction (see Figure 2). First, the system performs lightweight processing to identify movement and changes in the scene. Once such changes are detected, the CNN models verify the presence of key construction-related elements, such as workers, trucks, excavators, or other tools. If these elements are identified, irrelevant parts of the image, such as the background, are cropped out. A video segment with optimized resolution and frame rate is then sent to the cloud for further verification. Users are then notified accordingly.

3 DEMONSTRATION

We deployed EdgeCloudAI on the COSMOS testbed using its traffic cameras in NYC along with one of its edge servers equipped with an Nvidia A100 GPU. We used *gpt-4o* [7] as the large VLM in the cloud. EdgeCloudAI consumes between 5-7% of the edge GPU capacity. The average weekly cost of cloud querying does not exceed 10$ per camera when running 24/7. In this demonstration, we will showcase the real-time performance of EdgeCloudAI to detect user-defined incidents and notify the user accordingly.

ACKNOWLEDGMENT

This work was supported in part by NSF grants CNS-1652115, OAC-2029295, CNS-2148128, EEC-2133516, NSF grant CNS-2038984 and corresponding support from the Federal Highway Administration (FHA), ARO grants W911NF2210031 and W911NF1910379.

REFERENCES

[1] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. 2024. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247* (2024).

[2] Sandeep Chinchali, Evgenya Pergament, Manabu Nakanoya, Eyal Cidon, Edward Zhang, Dinesh Bharadia, Marco Pavone, and Sachin Katti. 2021. Sampling training data for continual learning between robots and the cloud. In *Proc. Springer ISER.*

[3] COSMOS Project. 2022. Hardware: Cameras. (2022). https://wiki.cosmos-lab.org/wiki/Hardware/Cameras.

[4] Kuntai Du, Qizheng Zhang, Anton Arapin, Haodong Wang, Zhengxu Xia, and Junchen Jiang. 2022. Accmpeg: Optimizing video encoding for accurate video analytics. In *Proc. MLSys.*

[5] Zixu Hao, Huiqiang Jiang, Shiqi Jiang, Ju Ren, and Ting Cao. 2024. Hybrid SLM and LLM for edge-cloud collaborative inference. In *Proc. EdgeFM.*

[6] Berthold KP Horn and Brian G Schunck. 1981. Determining optical flow. *Elsevier Artif, intell.* 17, 1-3 (1981), 185–203.

[7] Raisa Islam and Owana Marzia Moushi. 2024. GPT-4o: The cutting-edge advancement in multimodal LLM. *Authorea Preprints* (2024).

[8] Mehrdad Khani, Ganesh Ananthanarayanan, Kevin Hsieh, Junchen Jiang, Ravi Netravali, Yuanchao Shu, Mohammad Alizadeh, and Victor Bahl. 2023. RECL: Responsive resource-efficient continuous learning for video analytics. In *Proc. USENIX NSDI.*

[9] Seyeon Kim, Kyungmin Bin, Donggyu Yang, Sangtae Ha, Song Chong, and Kyunghan Lee. 2023. ENTRO: Tackling the encoding and networking trade-off in Offloaded video analytics. In *Proc. ACM MM.*

[10] Woo-Joong Kim and Chan-Hyun Youn. 2020. Lightweight online profiling-based configuration adaptation for video analytics system in edge computing. *IEEE Access* 8 (2020), 116881–116899.

[11] Yuxin Kong, Peng Yang, and Yan Cheng. 2023. Edge-assisted on-device model update for video analytics in adverse environments. In *Proc. ACM MM.*

[12] Man-Hee Lee, Hun-Woo Yoo, and Dong-Sik Jang. 2006. Video scene change detection using neural network: Improved ART2. *Elsevier Expert Syst. Appl.* 31, 1 (2006), 13–25.

[13] Zheng Lin, Guanqiao Qu, Qiyuan Chen, Xianhao Chen, Zhe Chen, and Kaibin Huang. 2023. Pushing large language models to the 6g edge: Vision, challenges, and opportunities. *arXiv preprint arXiv:2309.16739* (2023).

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *NeurIPS* 36 (2024).

[15] Weihong Liu, Jiawei Geng, Zongwei Zhu, Jing Cao, and Zirui Lian. 2022. Sniper: Cloud-edge collaborative inference scheduling with neural network similarity modeling. In *Proc. ACM/IEEE DAC.*

[16] Oscar Moll, Favyen Bastani, Sam Madden, Mike Stonebraker, Vijay Gadepally, and Tim Kraska. 2022. ExSample: Efficient searches on video repositories through adaptive sampling. In *Proc. IEEE ICDE.*

[17] Taslim Murad, Anh Nguyen, and Zhisheng Yan. 2022. DAO: Dynamic adaptive offloading for video analytics. In *Proc. ACM MM.*

[18] Sibendu Paul, Utsav Drolia, Y Charlie Hu, and Srimat T Chakradhar. 2021. Aqua: Analytical quality assessment for optimizing video analytics systems. In *Proc. IEEE/ACM SEC.*

[19] Dipankar Raychaudhuri, Ivan Seskar, Gil Zussman, Thanasis Korakis, Dan Kilper, Tingjun Chen, Jakub Kolodziejski, Michael Sherman, Zoran Kostic, Xiaoxiong Gu, et al. 2020. Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless. In *Proc. ACM MobiCom.*

[20] Neha Sharma, Vibhor Jain, and Anju Mishra. 2018. An analysis of convolutional neural networks for image classification. *Elsevier Procedia Comput. Sci.* 132 (2018), 377–384.

[21] Chris Stauffer and W Eric L Grimson. 1999. Adaptive background mixture models for real-time tracking. In *Proc. IEEE CVPR.*

[22] Kevin Swingler and Mandy Bath. 2020. Learning spatial relations with a standard convolutional neural network. In *NTCA.*

[23] Can Wang, Sheng Zhang, Yu Chen, Zhuzhong Qian, Jie Wu, and Mingjun Xiao. 2020. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics. In *Proc. IEEE INFOCOM.*

[24] Liang Wang, Kai Lu, Nan Zhang, Xiaoyang Qu, Jianzong Wang, Jiguang Wan, Guokuan Li, and Jing Xiao. 2023. Shoggoth: towards efficient edge-cloud collaborative real-time video inference via adaptive online learning. In *Proc. ACM/IEEE DAC.*

[25] Jianyu Wei, Ting Cao, Shijie Cao, Shiqi Jiang, Shaowei Fu, Mao Yang, Yanyong Zhang, and Yunxin Liu. 2023. NN-stretch: Automatic neural network branching for parallel inference on heterogeneous multi-processors. In *Proc. ACM MobiSys.*

[26] Zheming Yang, Wen Ji, Qi Guo, and Zhi Wang. 2023. JAVP: Joint-aware video processing with edge-cloud collaboration for DNN inference. In *Proc.ACM MM.*

[27] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. IEEE/CVF.*

[28] Mingjin Zhang, Jiannong Cao, Xiaoming Shen, and Zeyang Cui. 2024. EdgeShard: Efficient LLM inference via collaborative edge computing. *arXiv preprint arXiv:2405.14371* (2024).

[29] Wuyang Zhang, Zhezhi He, Luyang Liu, Zhenhua Jia, Yunxin Liu, Marco Gruteser, Dipankar Raychaudhuri, and Yanyong Zhang. 2021. Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading. In *Proc. ACM MobiCom.* 201–214.